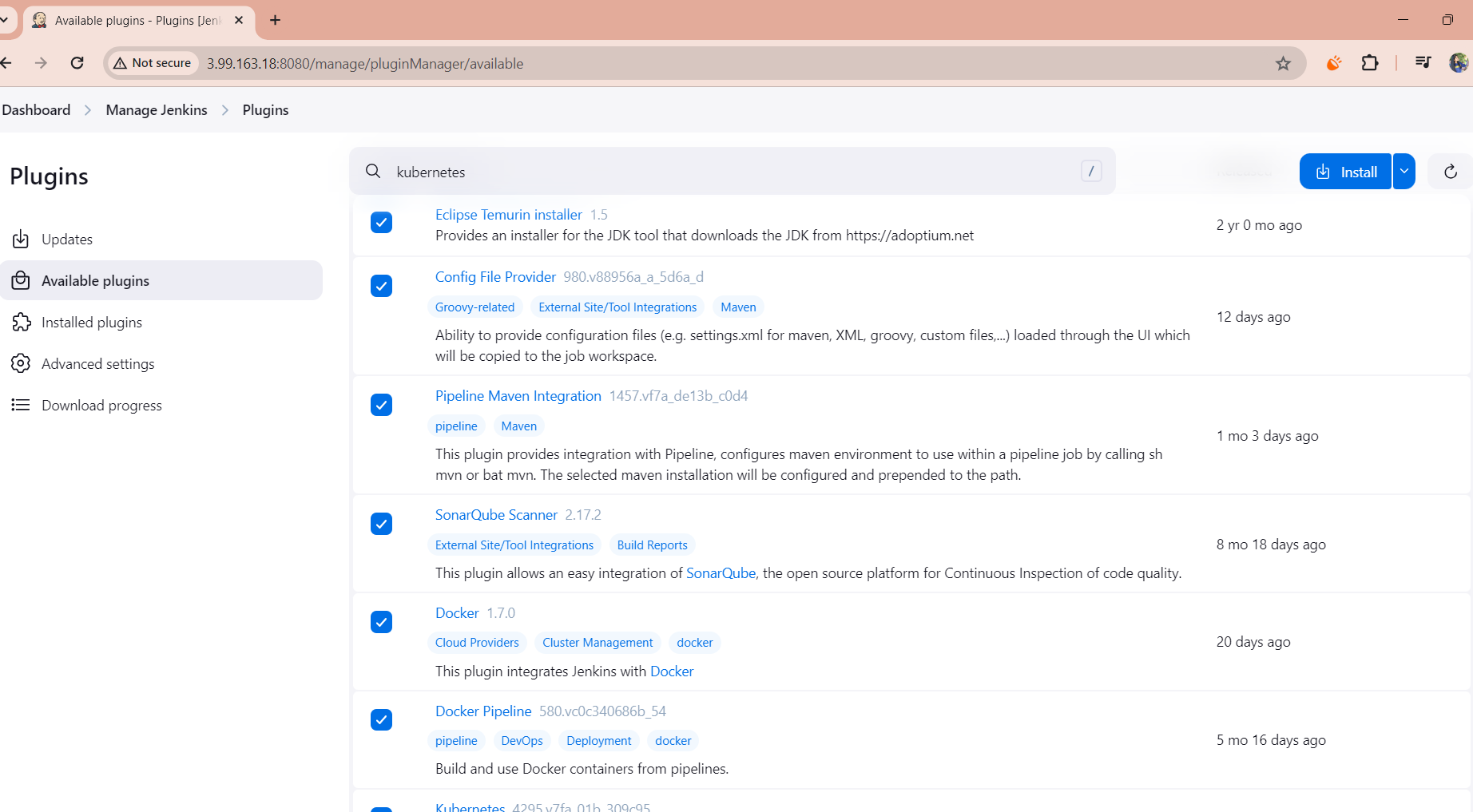
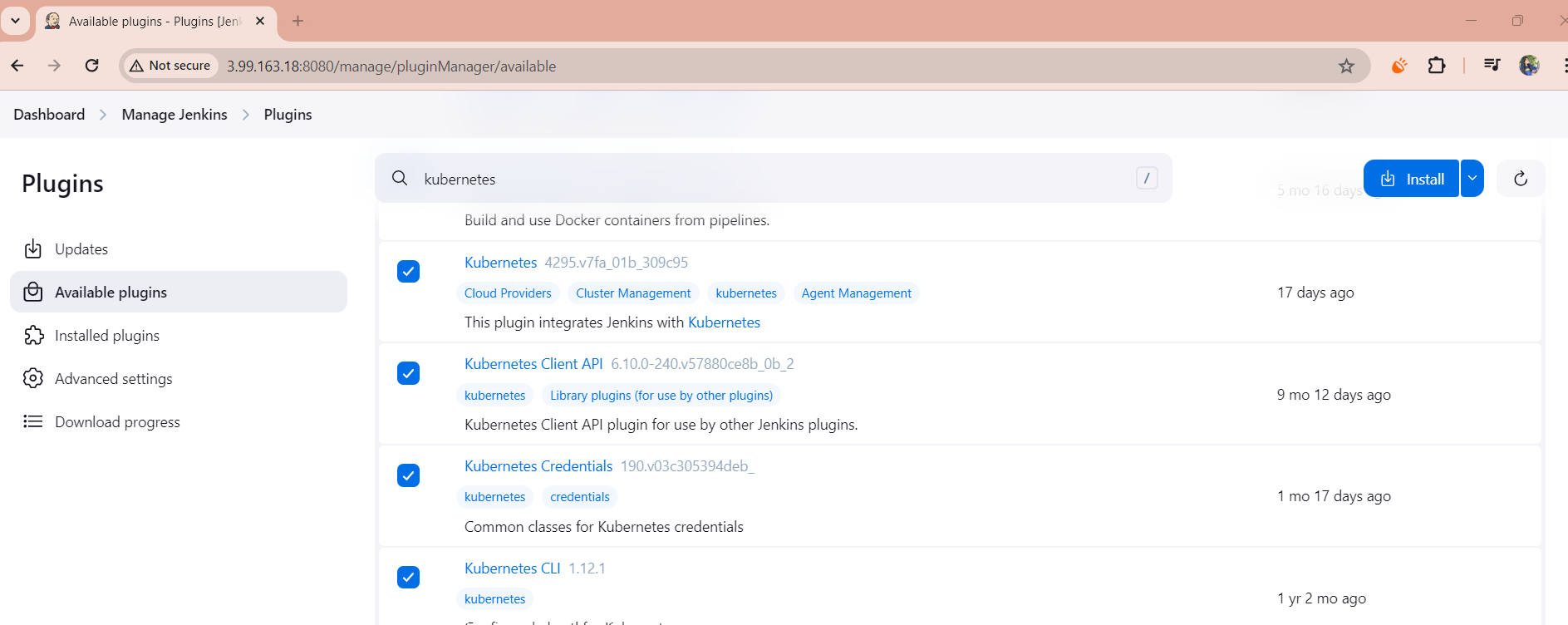
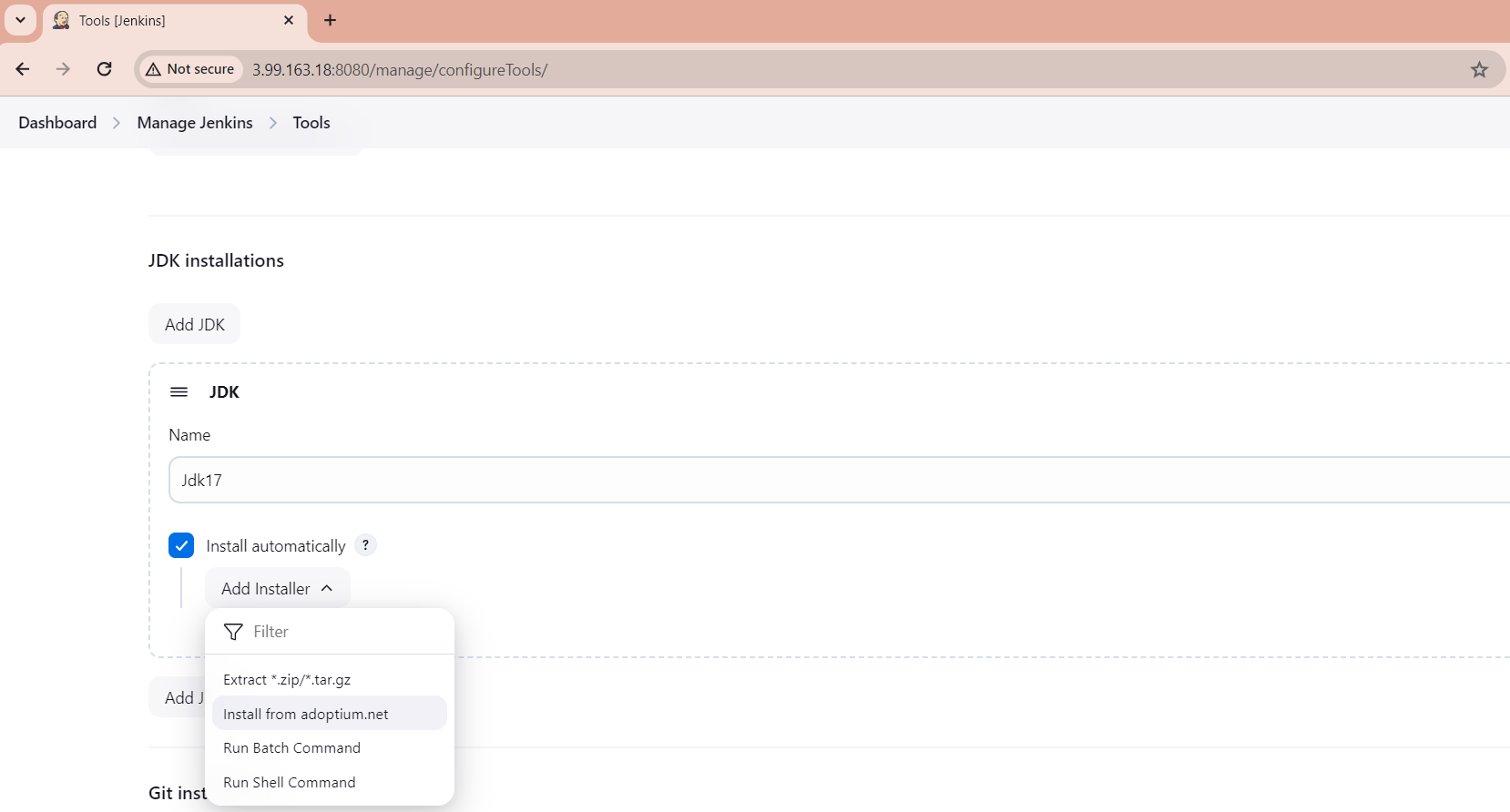
PHASE 3: CI/CD Pipelines

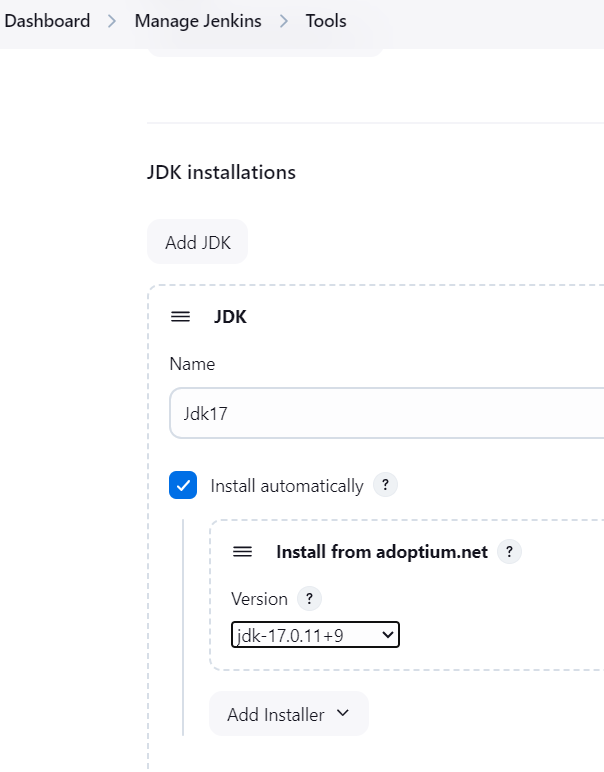
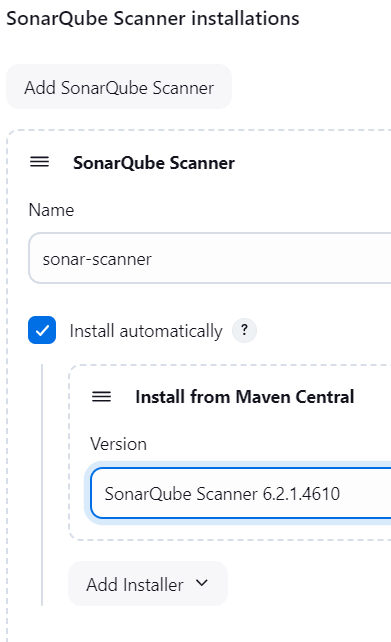
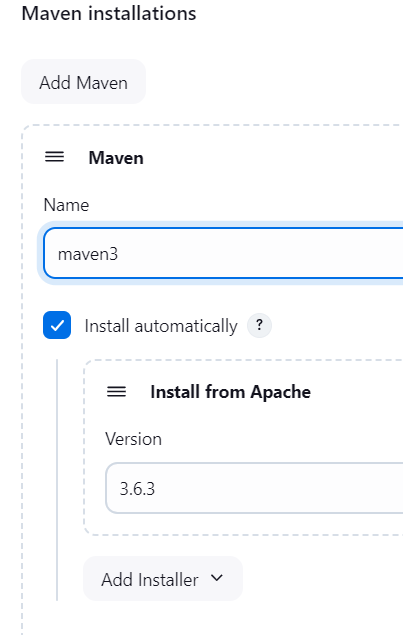
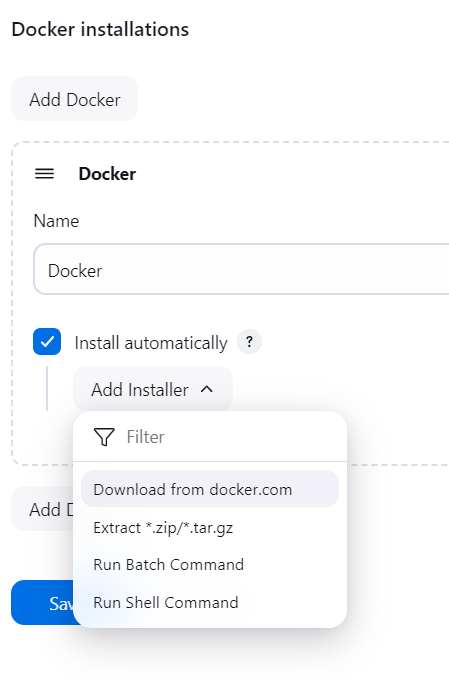
***Step1: Installing plugins on Jenkins***

* Go to Manage Jenkins- Plugins- Available Plugins
* Install following plugins:
* JDK
* Maven
* Maven Integration
* SonarQube Scanner
* Docker
* Kubernetes



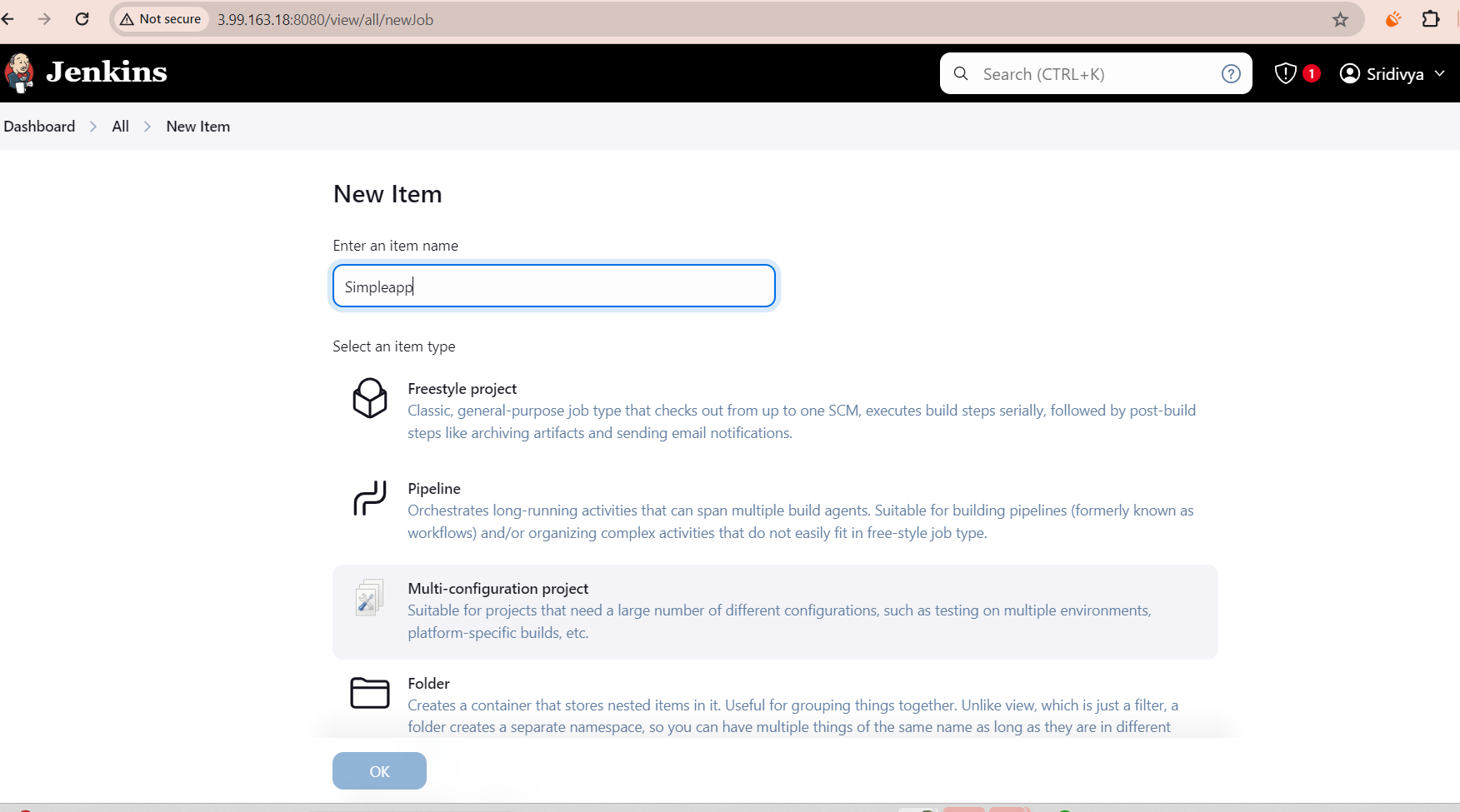
***Step 2: Configure Installed Plugins***

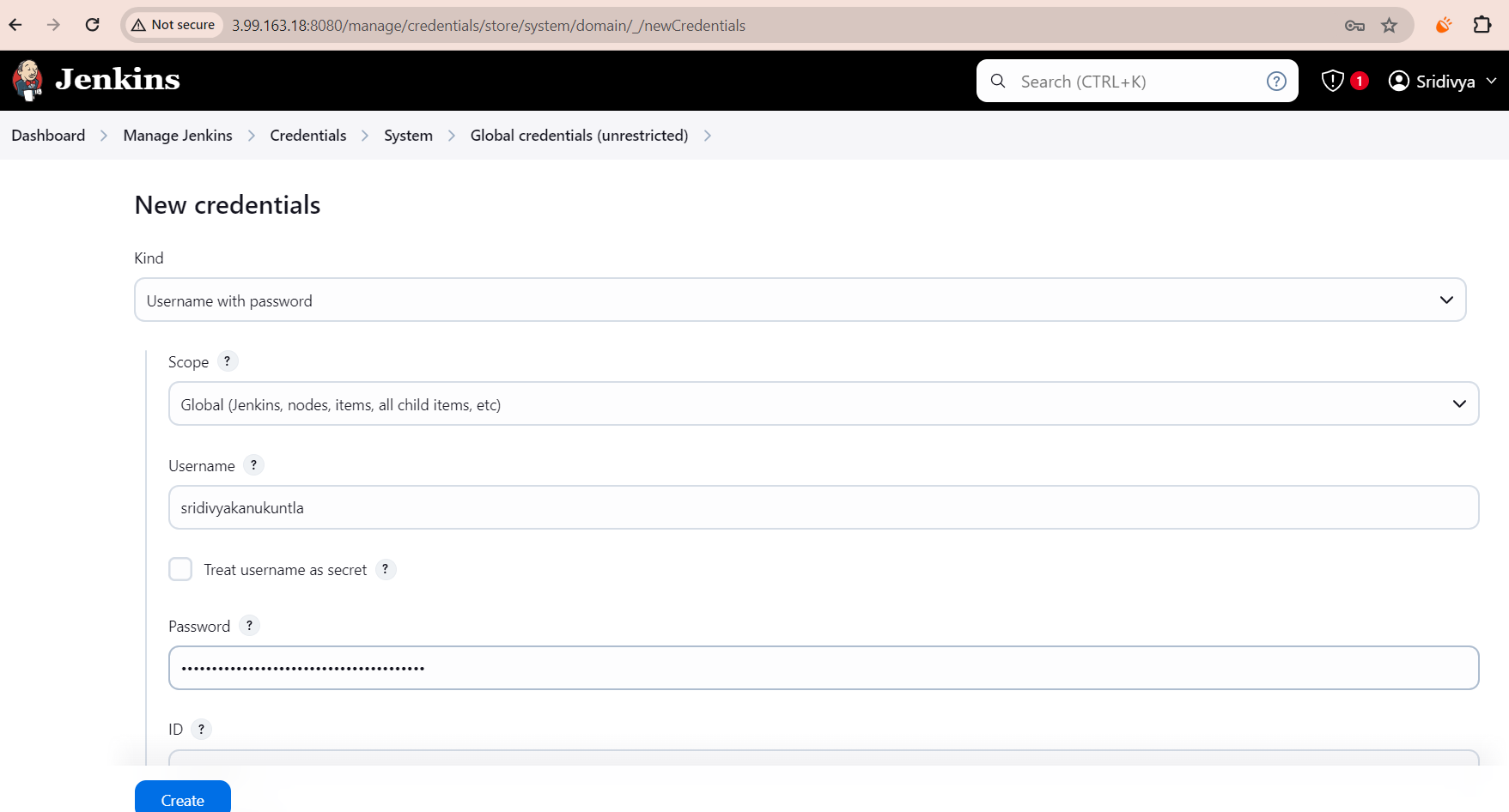
Go to Manage Jenkins > Global Tool Configuration.

Scroll to see the installed plugins and configure as needed:  

Now that the tools are configured, Let’s now create PIPELINE

***Step 3: Create Jenkins Pipeline***

* Go to New Item > Pipeline, give it a name (e.g., MySimpleAppPipeline), and click OK.
* In the configuration, choose the Hello World template, then edit it as shown below.
* Define tools (JDK and Maven) in the pipeline using the tools {} section.
* To set up Git credentials, create a GitHub token and add it under Manage Jenkins > Credentials.

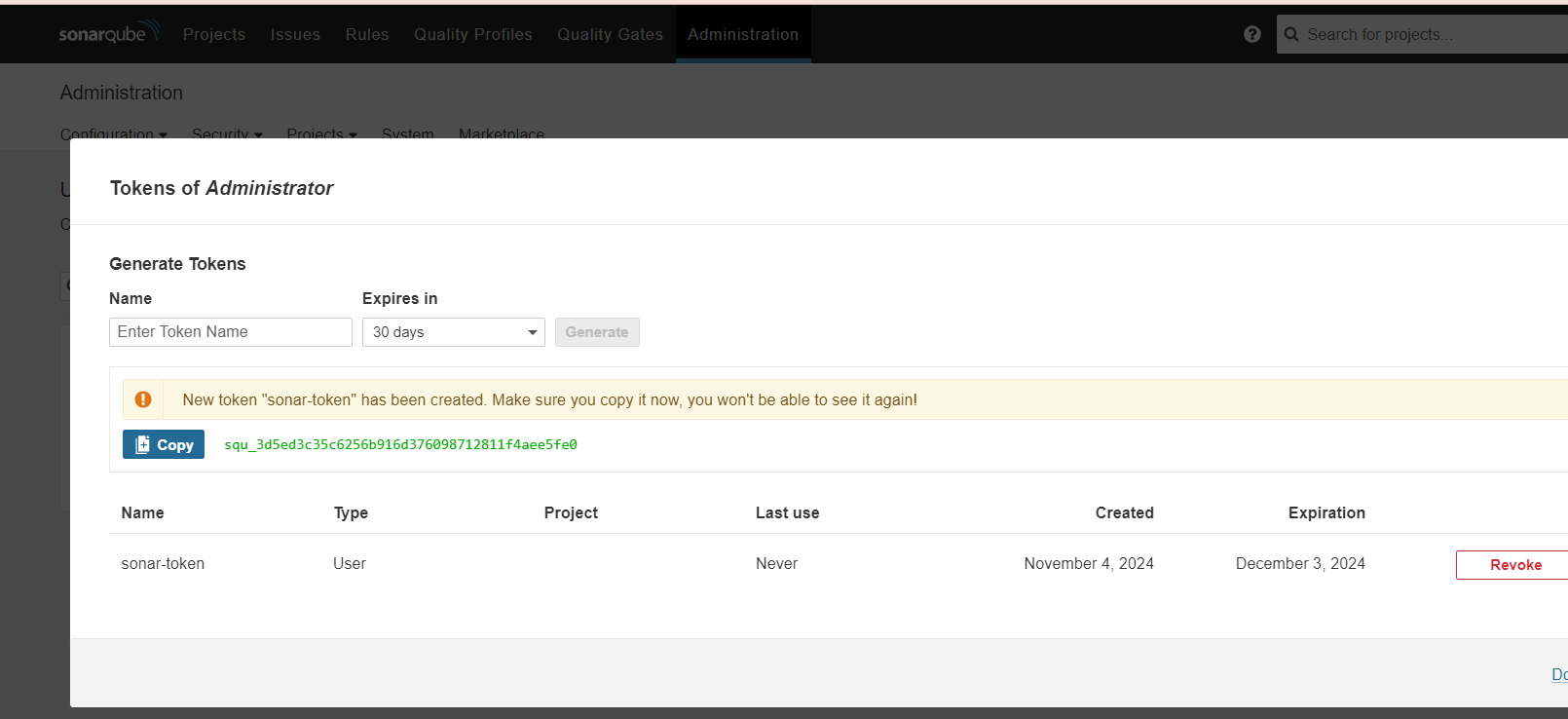
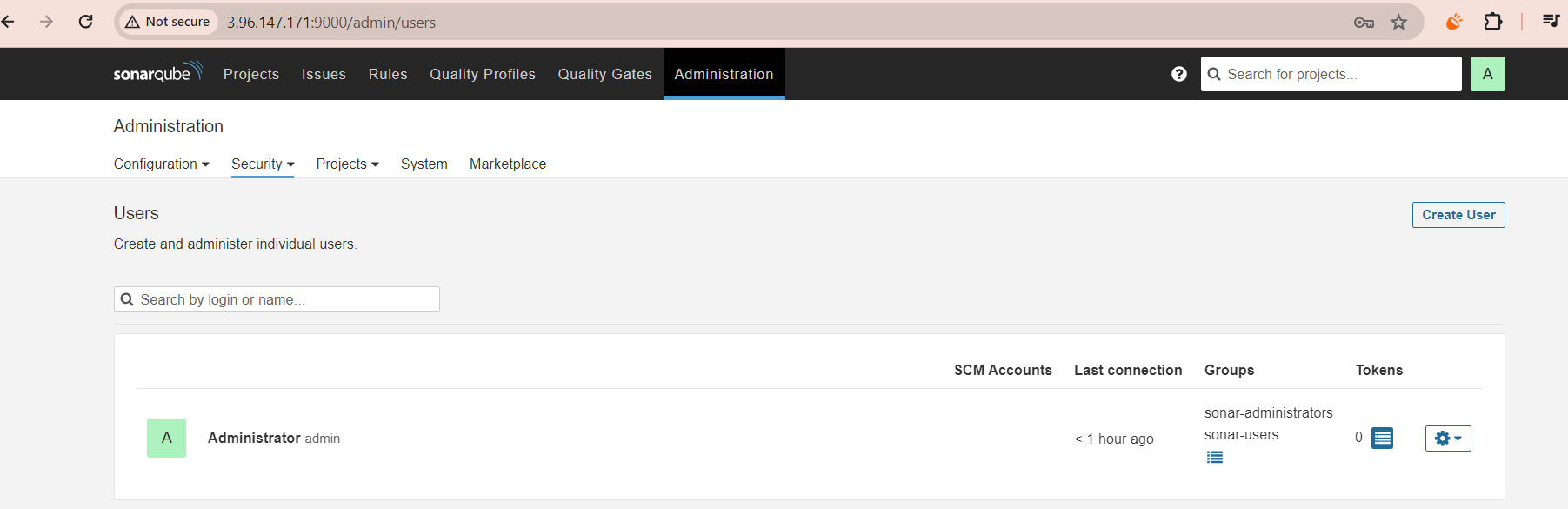


***Step 4: Installing Trivy on Jenkins Server***

* **Download Trivy** (replace 0.57.0 with the latest version if necessary):
  + wget https://github.com/aquasecurity/trivy/releases/latest/download/trivy\_0.57.0\_Linux-64bit.tar.gz
* **Extract the downloaded file**:
  + tar zxvf trivy\_0.57.0\_Linux-64bit.tar.gz
* **Move the binary to /usr/local/bin/**:
  + sudo mv trivy /usr/local/bin/
* **Make the Trivy binary executable**:
  + sudo chmod +x /usr/local/bin/trivy
* **Verify the installation**:
  + trivy –version

***Step 5: Setting up SonarQube Secret Token***

1. Go to SonarQube > Administration>Security >click on Tokens
2. Add the token in **Manage Jenkins** > **Credentials** to enable secure access to SonarQube.



#### **Step 6: Configure Webhook and Service Account for Kubernetes**

1. **Create a Webhook** in SonarQube: Go to **Admin** > **Webhooks** and set up a new webhook.
2. **Create a Service Account** for Jenkins in Kubernetes and apply the configurations:

root@ip-172-31-21-20:/home/ubuntu# vi svc.yaml

### Creating Service Account

apiVersion: v1

kind: ServiceAccount

metadata:

name: jenkins

namespace: webapps

root@ip-172-31-21-20:/home/ubuntu# **kubectl create ns webapps**

namespace/webapps created

root@ip-172-31-21-20:/home/ubuntu# **kubectl apply -f svc.yaml**

serviceaccount/jenkins created

root@ip-172-31-21-20:/home/ubuntu# vi role.yaml

### **Create Role**

apiVersion: rbac.authorization.k8s.io/v1

kind: Role

metadata:

name: app-role

namespace: webapps

**rules**:

- apiGroups:

- ""

- apps

- autoscaling

- batch

- extensions

- policy

- rbac.authorization.k8s.io

resources:

- pods

- secrets

- componentstatuses

- configmaps

- daemonsets

- deployments

- events

- endpoints

- horizontalpodautoscalers

- ingress

- jobs

- limitranges

- namespaces

- nodes

- pods

- persistentvolumes

- persistentvolumeclaims

- resourcequotas

- replicasets

- replicationcontrollers

- serviceaccounts

- services

**verbs**: ["get", "list", "watch", "create", "update", "patch", "delete"]

root@ip-172-31-21-20:/home/ubuntu**# kubectl apply -f role.yaml**

role.rbac.authorization.k8s.io/app-role created

root@ip-172-31-21-20:/home/ubuntu# vi bind.yaml

### Bind the role to service account

apiVersion: rbac.authorization.k8s.io/v1

kind: RoleBinding

metadata:

name: app-rolebinding

namespace: webapps

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: Role

name: app-role

subjects:

- namespace: webapps

kind: ServiceAccount

name: jenkins

root@ip-172-31-21-20:/home/ubuntu# **kubectl apply -f bind.yaml**

rolebinding.rbac.authorization.k8s.io/app-rolebinding created

**Create Service Account, Role & Assign that role, And create a secret for Service Account and generate a Token**

root@ip-172-31-21-20:/home/ubuntu# kubectl apply -f sec.yaml -n webapps

secret/mysecretname created

***Step 7: Install kubectl on Jenkins***

1. Download kubectl:

curl -o kubectl <https://amazon-eks.s3.us-west-2.amazonaws.com/1.19.6/2021-01-05/bin/linux/amd64/kubectl>

1. Make it executable and move to /usr/local/bin

chmod +x ./kubectl

sudo mv ./kubectl /usr/local/bin

1. Verify the installation:

kubectl version --short –client

***Step 8: Set Up Monitoring After Pipeline Execution***

Once the pipeline is built, configure monitoring for your services to track performance and detect issues. This is explained in Phase4.